Adversarial Penetration Analysis of an OS-Level Post-Quantum OTP-Like Key Fabric

Archive X scientific report for cryptography and systems security indexing
Time-to-decipher under compromise, MITM resistance, kernel nonce enforcement, and multi-region key contribution

Paul George Savluc

January 2026

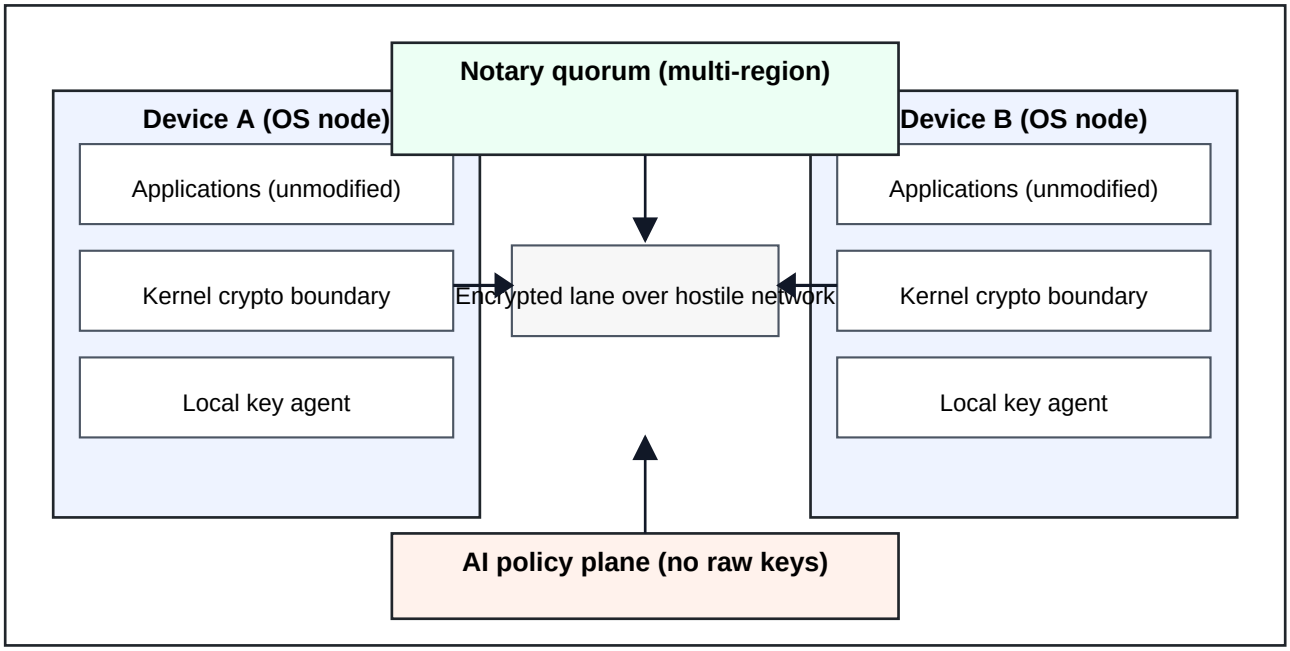


Figure 1. High-level OS key fabric architecture (conceptual).

SEO terms: post-quantum cryptography, ML-KEM, CRYSTALS-Kyber, OS-level encryption, kernel crypto boundary, OTP-like key rotation, forward secrecy, key fabric, network mesh encryption, zero trust networking, MITM resistance, device fleet security, notary quorum, key transparency, penetration analysis, breach impact modeling, time-to-decipher

Short abstract. Ciphertext-only attackers face infeasible deciphering at modern symmetric security levels. Meaningful deciphering usually requires endpoint compromise, governance collapse, or catastrophic correctness failure such as nonce reuse. This report quantifies deciphering time under each breach level and maps where security infringes in real networks.

Abstract and executive summary

This report evaluates an operating-system-level encryption fabric designed to approximate one-time-pad behavior through strict nonrepetition and high-rate rekeying, while using post-quantum key establishment (ML-KEM, standardized from CRYSTALS-Kyber) to prevent harvest-now-decrypt-later. The analysis is adversarial: we assume a hostile network, active man-in-the-middle attempts, partial infrastructure compromise, and endpoint-level breaches.

Ciphertext-only attackers face infeasible deciphering times at modern symmetric security levels. In contrast, breach-assisted attackers decipher immediately once they cross an endpoint boundary, so the real engineering metrics are exposure window, attacker dwell time, and blast radius. We evaluate a kernel interface contract that avoids long-lived kernel key state by installing only short-lived traffic keys, nonce rules, and replay parameters while keeping counters in-kernel to guarantee nonce uniqueness.

Safety note. This is a defensive audit. It does not provide step-by-step exploitation instructions.

Key contributions (summary):

- A breach taxonomy that separates cryptanalytic, protocol, and operational break modes.

- A time-to-decipher model with explicit assumptions and exposure-window calculations.

- A kernel contract review centered on epoch, anti-rollback, and nonce uniqueness invariants.

- A governance and multi-region notary analysis that resists re-centralization into a universal decryptor.

Contents

1. What it means to break the system
2. Threat model and attacker classes
3. System model and trust boundaries
4. Kernel interface contract review: short-lived keys and in-kernel counters
5. Time-to-decipher: compute baselines and breach-driven realities
6. Breach taxonomy and infringement locations across a networked fleet
7. Case studies: where security fails in real enterprise networks
8. Post-quantum bootstrap: ML-KEM, transcript binding, and key confirmation
9. Multi-region secret contribution: removing backend single points of decryption
10. Side channels, metadata leakage, and what OTP-like does not solve
11. Hardening roadmap and verification checklist
12. Conclusions
13. References

1. What it means to break the system

We separate three notions: cryptanalytic break, protocol break, and operational break. This prevents mixing math limits with system engineering realities.

Cryptanalytic break: plaintext recovery from ciphertext without endpoint secrets.

Protocol break: forcing the handshake into an insecure state (downgrade, parameter substitution, or missing transcript binding).

Operational break: obtaining keys or plaintext via endpoint, governance, or update-channel compromise.

2. Threat model and attacker classes

Assume the network is hostile and attackers can be active. Also assume realistic enterprise failures such as configuration drift, partial compromise, and supply chain risk.

A0 Passive recorder: records traffic and metadata, cannot alter messages.

A1 Active MITM: drops, delays, replays, injects, attempts downgrades, no endpoint control.

A2 Partial infrastructure compromise: controls some notary or relay nodes, not a threshold quorum.

A3 Governance compromise: influences membership or policy decisions, subverts monitoring.

A4 Endpoint agent compromise: controls the privileged local key agent on one device.

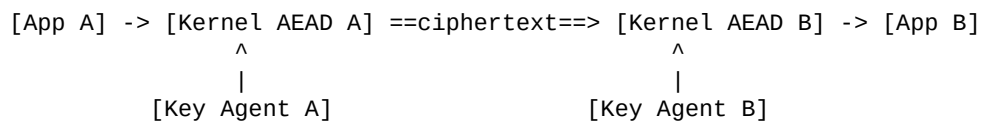
A5 Kernel boundary compromise: reads plaintext at the kernel datapath and can observe installed traffic keys.

A6 Hardware root compromise: misuses device identity operations, enabling impersonation.

3. System model and trust boundaries

Each node includes applications, a kernel crypto boundary, and a privileged local key agent. Applications remain unmodified. The kernel enforces AEAD, replay windows, and nonce uniqueness. The agent performs PQ bootstrap, ratcheting, and policy checks, then installs short-lived traffic keys into the kernel.

Figure 2. Data path over a hostile network (conceptual)



Attacker sees the ciphertext lane. Plaintext requires crossing an endpoint boundary.

4. Kernel interface contract review: short-lived keys and in-kernel counters

The kernel contract tries to make the kernel an enforcement boundary, not a long-lived key vault. This is good, but it is secure only if epoch and anti-rollback invariants are enforced.

```
Kernel interface contract (summary)
- Kernel receives only short-lived traffic keys, nonce derivation rules, and replay window parameters.
- Kernel maintains counters to guarantee nonce uniqueness.
- No long-lived master secret is stored in kernel.
```

Invariant 1: explicit epochs. A traffic key must be tied to an epoch identifier. Epochs must not repeat. If the kernel cannot prove the epoch is fresh after a reboot or crash, it must refuse to encrypt until rekeyed.

Invariant 2: anti-rollback for counters. Counters guarantee uniqueness only if they cannot roll back under the same epoch. If rollback is possible, nonce reuse can occur and can be catastrophic for affected epochs.

Invariant	Failure mode	If it fails	Control
Explicit epoch	Key reuse across reboot	Nonce uniqueness assumptions can break	Epoch change on reboot; monotonic enforcement
Anti-rollback counters	Counter reset under same epoch	AEAD misuse risk for that epoch	Persistent monotonic state or forced rekey
Kernel validates nonce rules	Rule substitution by compromised agent	Nonces may collide or be predictable	Transcript bind rule ID; signed rules
Key install gating	Unauthorized key injection	Attacker gains decrypt window	Capabilities; LSM/WFP-style policy; audit logs

5. Time-to-decipher: compute baselines and breach-driven realities

Time-to-decipher depends on which boundary fails. Ciphertext-only attackers face infeasible times at strong symmetric security levels. Breach-assisted attackers decipher immediately once live keys or plaintext are exposed.

Symmetric security (bits)	Time at 1e+12 guesses/sec	Interpretation
64	213.50 d	Legacy or borderline.
80	38,309 y	Legacy or borderline.
96	2.51e9 y	Strong classical margin.
112	1.65e14 y	Strong classical margin.
128	1.08e19 y	Modern baseline.
192	1.99e38 y	Huge margin.
256	3.67e57 y	Huge margin.

Table 1. Ciphertext-only brute force scale intuition at 1e+12 guesses per second.

Symmetric security (bits)	Time at 1e+15 guesses/sec	Interpretation
64	5.12 h	Legacy or borderline.
80	38 y	Legacy or borderline.
96	2.51e6 y	Strong classical margin.
112	1.65e11 y	Strong classical margin.
128	1.08e16 y	Modern baseline.
192	1.99e35 y	Huge margin.
256	3.67e54 y	Huge margin.

Table 1. Ciphertext-only brute force scale intuition at 1e+15 guesses per second.

Symmetric security (bits)	Time at 1e+18 guesses/sec	Interpretation
64	18.45 s	Legacy or borderline.
80	13.99 d	Legacy or borderline.
96	2,511 y	Strong classical margin.
112	1.65e8 y	Strong classical margin.
128	1.08e13 y	Modern baseline.
192	1.99e32 y	Huge margin.
256	3.67e51 y	Huge margin.

Table 1. Ciphertext-only brute force scale intuition at 1e+18 guesses per second.

5.1 Exposure window model

Once an attacker reaches the agent or kernel boundary, deciphering is immediate. The defense becomes: rotate fast (small L), detect fast (small D), quarantine fast (small Q).

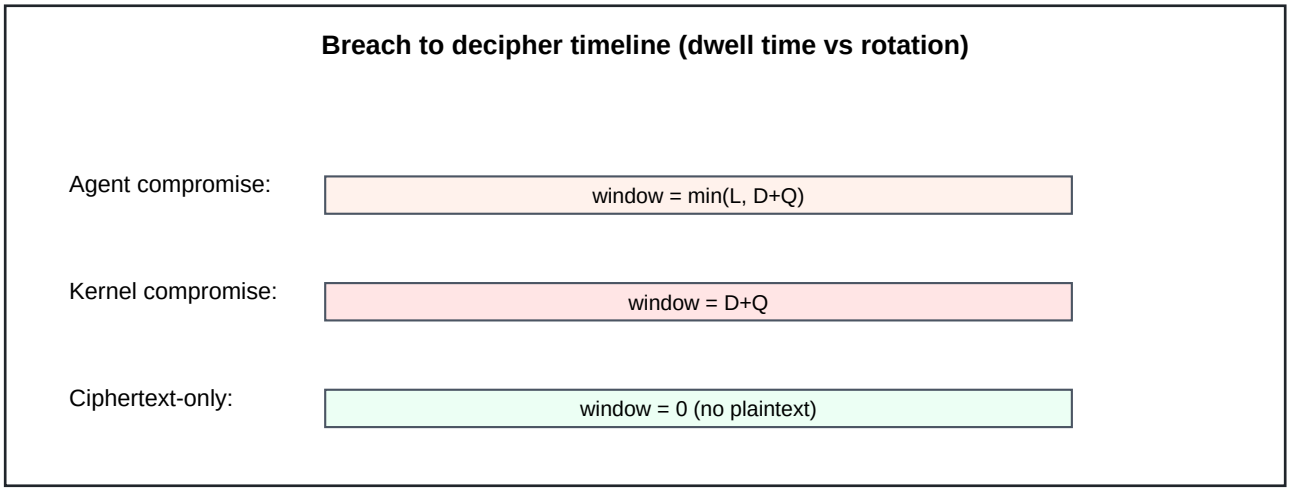


Figure 3. Exposure window depends on rotation, detection, and quarantine.

Scenario	L (rotation)	D (detection)	Q (quarantine)	Agent window	Kernel window
Tight security	1.00 s	5.00 min	1.00 min	1.00 s	6.00 min
Good SOC	10.00 s	30.00 min	5.00 min	10.00 s	35.00 min
Typical enterprise	1.00 min	4.00 h	10.00 min	1.00 min	4.17 h
Weak monitoring	10.00 min	24.00 h	1.00 h	10.00 min	25.00 h

Table 2. Example exposure windows under four operational postures. L is key rotation. D is detection. Q is quarantine.

6. Breach taxonomy and infringement locations across a networked fleet

This taxonomy maps compromise levels to what leaks and where the infringement occurs. It is the practical map for incident response planning.

Breach level	What attacker gets	Time-to-decipher	Typical exposure window	Dominant control
0 Passive	Ciphertext + metadata	Infeasible	0	AEAD; authentication; nonces
1 MITM only	Tamper/replay; downgrade attempts	Cannot decrypt; should fail loud	0	Transcript binding; key confirmation
2 Single notary	One component secret	Infeasible if threshold combiner used	0	Threshold governance; transparency
3 Governance drift	Future policy influence	Future sessions at risk	Until corrected	Multi-party approval; logs; audits
4 Agent compromise	Current ratchet state or traffic keys	Immediate	min(L, D + Q)	Rotate L; detect D; quarantine Q
5 Kernel compromise	Plaintext in datapath; installed keys	Immediate	D + Q	Hardening; isolation; EDR
6 Hardware root	Identity operations; impersonation potential	Immediate for impersonation	Until revoked	TPM/HSM; revocation
7 Nonce/epoch bug	AEAD assumptions violated	Fast on affected epoch	Epoch-dependent	Nonce discipline; anti-rollback

```
Figure 4. Typical infringement locations
- Passive taps: ISP, WAN, backbone, WiFi, cables.
- Active MITM: compromised routers, proxies, BGP or DNS interference, malicious WiFi.
- Agent: local privilege escalation, credential theft, poisoned update, debugger abuse.
- Kernel: kernel exploits, malicious modules, DMA, hypervisor escape.
- Governance: quorum shrinkage, policy tampering, transparency bypass.
```

7. Case studies: where security fails in real enterprise networks

7.1 Downgrade and parameter substitution

If suite selection, policy version, notary set, nonce-rule ID, and key confirmation are not transcript-bound, a MITM can steer sessions into weaker modes or confuse peers. Defense: strict transcript binding and explicit key confirmation.

7.2 Counter rollback after reboot

Counters are only helpful if they cannot roll back under the same epoch. Reboot storms create ambiguity. Defense: epoch change on reboot or monotonic counters, and refusal to encrypt until rekeyed when uncertain.

7.3 Fleet-wide update poisoning

A compromised update channel defeats cryptography by distributing malicious agents or kernel modules. Defense: signed updates, reproducible builds, staged rollouts, and independent verification.

7.4 Policy plane compromise

If the policy plane can unilaterally change quorum membership or disable authentication, the system becomes a centralized decryptor. Defense: multi-party control, transparency logs, and kernel-enforced baselines.

7.5 Endpoint memory disclosure and DMA

Short-lived keys still reside in RAM. DMA and kernel memory disclosures can reveal live keys. Defense: IOMMU, isolation, and rapid quarantine.

8. Post-quantum bootstrap: ML-KEM, transcript binding, and key confirmation

ML-KEM establishes shared secrets without sending them over the wire, supporting resistance to future quantum adversaries. It must be combined with identity authentication, transcript binding, and explicit key confirmation to resist MITM.

```
Minimal handshake safety checklist
1) Authenticate identities (certificate or hardware-bound identity)
2) Transcript-bind all parameters (suite, policy, notary set, nonce-rule ID)
3) Key confirmation (proof both sides derived the same secret)
4) Derive traffic keys via a KDF with domain separation
5) Install keys into kernel with explicit epoch, replay window, nonce rules
```

9. Multi-region secret contribution: removing backend single points of decryption

Multi-region contribution reduces risk of a single backend becoming a universal decryptor. Endpoints request independent contributions from notary sets in different regions and combine them so that missing required shares prevents reconstruction.

This helps only if the combiner is threshold-based and governance prevents quietly reducing the threshold. Membership and thresholds should be logged and monitored.

```
Threshold contribution idea (high level)
- Independent contributions from notaries in multiple regions
- Threshold combiner requires multiple shares
- Transparency logs for membership and threshold parameters
- Multi-party approval for critical changes
```

10. Side channels, metadata leakage, and what OTP-like does not solve

OTP-like nonrepetition addresses keystream reuse failures. It does not hide metadata and does not prevent endpoint compromise.

Metadata leakage: packet sizes, timing, and endpoints. Mitigations include padding, batching, and cover traffic, with cost tradeoffs.

Side channels: timing, cache, branch predictors, power, and electromagnetic leakage. Mitigations include constant-time implementations and hardware isolation.

Endpoint compromise: owned endpoints read plaintext. Mitigations focus on minimizing exposure window and isolating blast radius.

11. Hardening roadmap and verification checklist

The strongest cryptography fails when correctness and governance fail. Prioritize these controls.

- Kernel enforces AEAD and owns nonce uniqueness.
- Explicit epochs with monotonic behavior and anti-rollback handling.
- Transcript binding for suite, policy version, notary set, nonce-rule ID, and key confirmation.
- Directional key separation and KDF domain separation.
- Zeroization and secret-memory discipline in agent and kernel paths.
- Non-exportable hardware identities and revocation support.
- Transparency logs with independent monitors.
- Multi-party approval for critical policy changes; AI is capability-limited.
- Signed updates and staged rollouts with audit trails.
- Telemetry to reduce detection time D and quarantine propagation time Q.

12. Conclusions

Ciphertext-only deciphering is infeasible when the handshake is authenticated and keys are rotated with strict nonce discipline. The largest practical risks are operational: endpoint compromise, update-channel compromise, governance drift, and correctness failures such as nonce reuse or counter rollback.

The kernel interface contract is secure only when epochs and anti-rollback rules are enforced. Short-lived keys reduce blast radius but do not remove the need for hardening and rapid detection.

Signed,

Paul George Savluc

13. References

- NIST. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM). NIST CSRC. Effective date published in Federal Register: August 14, 2024.

- NIST. FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA). NIST CSRC.

- IETF. RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3. August 2018.

- IETF. RFC 5869: HMAC-based Extract-and-Expand Key Derivation Function (HKDF). 2010.

- IETF. RFC 8439: ChaCha20 and Poly1305 for IETF Protocols. 2018.

- IETF. Internet-Draft: ML-KEM Security Considerations. 2025.